

ON THE HITCHHIKER ROBOT OPERATED MATERIALS PROCESSING SYSTEM EXPERIMENT DATA SYSTEM

Semion Kizhner and Del Jenstrom
NASA Goddard Space Flight Center, Greenbelt MD

ABSTRACT

The Space Shuttle Discovery STS-64 mission carried the first American autonomous robot into space, the Robot Operated Materials Processing System (ROMPS). On this mission ROMPS was the only Hitchhiker experiment and had a unique opportunity to utilize all Hitchhiker space carrier capabilities. ROMPS conducted rapid thermal processing of one hundred semiconductor material samples to study how micro gravity affects the resulting material properties. The experiment was designed, built and operated by a small GSFC team in cooperation with industry and university based principal investigators who provided the material samples and data interpretation. ROMPS' success presents some valuable lessons in such cooperation, as well as in the utilization of the Hitchhiker carrier for complex applications. The motivation of this paper is to share these lessons with the scientific community interested in attached payload experiments. ROMPS has a versatile and intelligent material processing control data system. This paper uses the ROMPS data system as the guiding thread to present the ROMPS mission experience. It presents an overview of the ROMPS experiment followed by considerations of the flight and ground data subsystems and their architecture, data products generation during mission operations, and post mission data utilization. It then presents the lessons learned from the development and operation of the ROMPS data system as well as those learned during post-flight data processing.

ROMPS OVERVIEW

The Space Shuttle Discovery STS-64 mission carried the first American autonomous robot into space, ROMPS, in September of 1994. The ROMPS Hitchhiker experiment goal was to demonstrate commercial methods of processing semiconductor materials in a microgravity environment, namely rapid thermal processing of 100 thin film semiconductor material samples in space to improve the properties of the materials. The basis for this experiment is the fact that the performance of most semiconductor materials is dependent on their crystalline structure. Gravity driven convection and sedimentation, which disturb crystal formation, can be eliminated in the micro gravity of space. The crystal structures of material samples are reformed by ROMPS in heating and cooling cycles. The material processing was facilitated by an autonomous 4-degrees of freedom cylindrical robot (ref. 1).

ROMPS required two Space Shuttle sidewall mounted Get Away Special(GAS) cans, one containing the two furnaces, material samples, the robot, the furnace controller, and drive electronics; the other containing control electronics. Both cans were backfilled with one atmosphere of dry nitrogen to allow for the use of commercial grade and other non-vacuum rated components. The Hitchhiker avionics system provided ROMPS with power and with the communication channels for commands and telemetry.

The thermal-time processing profile for each material sample was parameterized by the Principal Investigators. These parameters were uploaded during the mission. The power to the halogen lamp parabolic focusing furnace was servoed using the parameters to cause the prescribed thermal-time profile at the furnace focus where a material sample is positioned by the robot. The material samples were housed in sample holders on individual pallets. The pallets were stored in six racks each capable of holding approximately 25 pallets. The job of sample pallet removal from a rack, placing the pallet into the furnace, holding for the duration of the

sample thermal-time profile, removal from furnace and returning to the rack was performed by the ROMPS robot.

ROMPS had several calibration samples representing each of the nine types of material samples. All calibration samples were processed first and actual temperature-time profiles were observed at the Payload Operations Control Center (POCC) using real time telemetry. The flight calibration data was compared with laboratory data obtained preflight. A calibration sample pallet has two thermocouples which contact the furnace spring leads when the sample is placed and held in contact with the furnace by the robot for the duration of processing. Thus the actual temperature was measured and telemetered to the POCC. The processing temperature profiles could then be adjusted by ground commands in the event that furnace lamp performance varied from preflight measurements.

ROMPS autonomous processing was organized into small sets of material samples. Each set was preceded and followed by processing of a corresponding material calibration sample to provide a frequent record of furnace performance. The processing of a set of material samples was initiated and controlled by ground commands, and telemetry was used in real time to monitor progress. The ROMPS experiment was capable of processing the entire compliment of material samples on-board without ground intervention once initiated. However, autonomously processing only small sets of samples at a time afforded several benefits; 1) processing during the frequent predictable telemetry outages could be avoided, 2) experiment performance could be analyzed prior to proceeding to the next set of samples, and 3) it was insured that processing took place only during designated ROMPS experiment operating periods. The processed samples were removed from the experiment after landing and studied at Principal Investigators (PI) laboratories to determine any resulting beneficial properties.

An experiment called the Capaciflector proximity sensor was added to the ROMPS robot for a flight demonstration of this sensing technology. After all material processing activities had been completed, the Capaciflector sensor was successfully used to demonstrate sensor-based robot control by guiding the robot to align and mate a material pallet to the processing furnace. Performance was monitored and verified by using the robot's own optical position encoders.

The entire ROMPS experiment and mission were facilitated and controlled by the ROMPS data system. The data system was developed primarily by the Space Automation and Robotics Center (SpARC) at the Environmental Research Institute of Michigan (ERIM), a NASA Center for the Commercial Development of Space (NASA CCDS), in conjunction with its commercial partners Interface and Control Systems, Inc. (ICS) and Zymark Corporation, under contract to the ROMPS project. Other portions of the data system were developed by ITE, Inc. of Laurel, MD., or they were developed in house at the NASA Goddard Space Flight Center (GSFC). The ROMPS data system consideration is used in this paper as the guiding thread to describe the ROMPS mission experience.

DATA SYSTEM DESIGN METHODOLOGY

In keeping with the commercial orientation and goals of the ROMPS mission and the ERIM/SpARC NASA CCDS, the ROMPS data system was built around a number of commercially available products. This greatly influenced the design that emerged. Combining a commercial spacecraft control software package along with a commercial robot controller eliminated much of the low level system design and development typically required to build a control system, and resulted in a highly capable and flexible system. However, significant effort was expended tackling compatibility, interface, and reliability issues, and resulted in a hierarchical design of significant complexity.

Requirements for the data system were functionally partitioned into several groups for implementation - experiment control computer requirements, robot controller requirements, furnace controller requirements, and ground station requirements (ref. 2, 3). Functional partitioning is a contemporary structured design methodology. This methodology, along with current availability of commercial software and hardware technologies that can support this design methodology, allowed fast prototyping and development of the ROMPS data system. This approach also provided flexibility. For example, a few months before delivery of the ROMPS experiment a new experiment, the Capaciflector, was added to the configuration. The same methodology was used to rapidly change the ROMPS data system to include the Capaciflector telemetry and commands.

Significant use was made of commercial electronic boards. All commercial boards were properly supported, heat sunk, staked, and coated, using NHB-5300 as a guideline (ref. 6). Custom made boards used military grade parts where available. All hardware and software were thoroughly tested, including environmentally testing each electronic box prior to integrated testing with the rest of the experiment.

For reliability and safety reasons, each processor was equipped with a watchdog timer that would reboot the processor into a benign state should its software fail. A hardware watchdog override was also installed using one of the Hitchhiker discrete commands in the event that a watchdog timer failed, preventing operation of the system. In addition, due to safety concerns with the furnace, temperatures and lamp on-time were monitored in several different computers, and hardware/software interlocks were designed to prevent inadvertent lamp operation.

A Hitchhiker Carrier hardware emulator was developed by the ROMPS project. The emulator was configured in a single mobile rack and was used at different sites at GSFC and the Kennedy Space Center (KSC) for ROMPS payload development, integration and test (I&T), and post-mission payload checkout.

FLIGHT SYSTEM ARCHITECTURE OVERVIEW

The flight data system consists of 4 computers as shown in Figure 1. Each are described below.

Spacecraft Command Language Computer (SCC)

Hardware: Single Board 80286 equivalent 10MHz CPU, A/D converter and multiplexer boards.

Function: SCC ran a VERTEX operating system and two tasks, DataIO and Spacecraft Command Language (SCL), both commercial products of Interface and Control Systems, Inc. (ref. 4). The DataIO and SCL tasks were used as a command and telemetry interface to Hitchhiker avionics over an RS-422 1200 baud interface. The ROMPS telemetry downlink originated from 150 telemetry data sources at 1 or 30 second intervals. The SCC collected 48 discrete inputs, about 50 analog inputs, and actuated 16 discrete commands. It also derived many telemetry points that reported process parameters and status. SCC commanded the ROMPS robot by executing stored scripts or forwarding ground serial asynchronous commands. Commands to the robot were sent to the EasyLab Computer over an RS-232-C 1200 baud serial asynchronous interface.

EasyLab Computer (EZC)

Hardware: Two board 80386 equivalent 20MHz CPU and quad serial board.

Function: EZC ran the EasyLab System V Controller software, a commercial product of Zymark Corporation, which stored the high level functions and positions for the ROMPS robot

and furnace controller. It effected robot moves by sending commands to the XPC Controller over an RS-232-C 1200 baud interface. It effected furnace operation by sending furnace commands to the furnace controller over an RS-422 interface, and a single bi-level furnace enable control signal.

XP Computer (XPC)

Hardware: Modified and repackaged Zymark Corporation XP controller based on 80C188 processor.

Function: XPC operated as a slave device to the above described EZC over its RS-232-C 1200 baud interface. The XPC provided low level servo control functions for the ROMPS robot.

Furnace Controller

Hardware: 80C188 controller with assorted custom electronics.

Function: The furnace controller was connected to the EZC over an RS-422 interface to receive serial asynchronous commands that specify the temperature profile. It was also connected to EZC by a single bi-level voltage line. When this line was commanded high the furnace was turned ON and the thermal profile stored in the furnace controller was started; when the line was commanded low the furnace was turned OFF.

FLIGHT SOFTWARE

SCC software was developed on a PC and tested on the developers SCC prototype card. The tested flight software was loaded into flight system RAM for testing and flight PROMs for I&T completion and the mission. The EZC was programmed on the ground development system prototype and loaded into EZC from SCC for ground tests. For flight configurations EZC PROMS were prepared and installed. The SCC ran the on-board SCL task with its database and the SCL run time engine for processing ROMPS scripts and rules. It also ran the DataIO task for data commutation and downlink, received commands from the HH avionics, and distributed the commands throughout the ROMPS flight data system (ref. 3). The software was developed incrementally and organized in five deliveries. The data system development was placed under strict configuration control three months prior to delivery of the payload to KSC, and frozen approximately 2 months prior to delivery.

FLIGHT SYSTEM/GROUND SYSTEM COMMUNICATIONS

The telemetry and command rates were both 1200 baud. The experiment telemetry frame was 110- bytes long and the telemetry frame cycle was 1 second. The experiment housekeeping data frame was 50-bytes long on a 30-second cycle. Flight-ground dialog text messages were down linked in telemetry frames of variable length. During the ROMPS mission 90 Mbytes of telemetry were collected and 1600 command blocks were uplinked. The command and telemetry communication channels were provided by the Hitchhiker project.

GROUND SYSTEM ARCHITECTURE OVERVIEW

The ground system requirements could most likely have been met using one powerful workstation. However, the ground system was implemented as a distributed system using several Macintosh computers. This distributed system was chosen for reasons of cost and compatibility with Spacecraft Command Language software. Its features included fast Ethernet interface of the computers into a 4-computer network, a copy of the ROMPS database on each computer, and database synchronization over the network. The primary ground system architecture is shown in Figure 2.

Once the primary ground system was integrated and tested in the POCC approximately 6 months prior to launch, the need to control that configuration required that it not be removed for travel to KSC for shuttle integration activities. Recognizing that this situation would occur, a second 2-computer ground system was developed that was used for activities at KSC. In addition, this system was used as a backup during flight operations. In parallel with the primary system, it received and archived ROMPS telemetry, and was available for payload commanding should the primary system go down.

GROUND SOFTWARE

The four computers of the primary ground system were labeled RGSEA1 through RGSEA4. RGSEA1 ran the DataIO task. This task received telemetry from the HH Customer/Carrier Ground Support Equipment (CCGSE) and displayed the flight system dialog messages on the computer screen. It searched the raw telemetry stream for the ROMPS telemetry frame synchronization pattern (SYNC) and, when SYNC was found, DataIO checked the frame Cyclic Redundancy Code(CRC). When both the SYNC and CRC were correct, DataIO archived the telemetry frame to the archive data set. DataIO then updated the computer database and sent database updates to two of the downstream computers, the Display computer RGSEA2 and the Command computer RGSEA3. DataIO also encoded the symbolic commands received from the RGSEA3 Command computer into HH-format commands, wrote them to the archive file and then sent them to the CCGSE.

The Display computer RGSEA2 ran a DataIO task to receive database updates from RGSEA1 and synchronize its database with RGSEA1, and a LabVIEW task to graphically display ROMPS telemetry. The LabVIEW task retrieved items from the database using a specially developed SCL interface task.

The Command computer RGSEA3 ran the ground SCL task with its database and the SCL intelligent expert system run time engine for processing ROMPS scripts and rules. These were ground equivalents of corresponding flight software tasks. RGSEA3 also ran the SCL ROMPS Project Version 3.0. and the DataIO task to update its database. Commands to the flight experiment entered at this terminal were automatically forwarded by DataIO to RGSEA1 for reformatting and transmission to the HH CCGSE.

The RGSEA4 computer was used for off-line data processing. Updates to material processing parameter tables and processing schedule scripts were developed on RGSEA4. These updates were copied to the command computer through the 4-computer Ethernet for uploading to the flight system. RGSEA4 was also used for near-real-time processing of data products such as generating graphs of furnace performance.

OPERATIONAL PROCEDURES AND DATA ARCHIVES

Thorough documentation is invaluable for maintaining control of and insuring efficient use of a data system. A complete set of ROMPS documentation was planned and systematically developed including software operators guides, a ground system user's guide, and a data products generation guide. These documents were used to develop operations procedures and served as main reference documents for ROMPS experiment I&T, flight operations, and post mission data processing.

ROMPS experiment operations were structured and controlled by developing a set of ROMPS operational procedures for each of four areas of ground operations competence: principal investigator (the team leader of a given ROMPS operating shift), lead operator (the single person on a given operating shift responsible for every command sent to the payload), operator (assistant to the lead operator in monitoring payload health and command generation),

and technical support (science principle investigators and other personnel with specific technical competence). In the lead operator's procedures, every command planned for the nominal mission was listed in order along with an estimated time required to implement each command. This proved to be invaluable for insuring that the proper commands got sent in the proper order, especially during the first few days of the mission when everyone was excited. This list was also very useful for planning which commands could be executed within each command window allocated to the ROMPS experiment throughout the mission.

In addition to individually outlining the responsibilities of each of the four positions, these procedures required that each individual manually record all significant events in a logbook designated to each position. These logbooks have provided redundant written records of mission events that have proven to be extremely useful in archive data interpretation.

The archive data sets automatically generated by the RGSEA1 DataIO task consisted of symbolic command log files (92K bites), ASCII files of dialog messages from the flight system (2.4 M bytes), and the main telemetry data set (approximately 90 M bytes). Similar archives of these last two data sets were also compiled by the backup ground system. The backup system did not archive commands since no commands were sent from it.

During the mission, the archived data sets were used to produce graphs of calibration sample temperature versus time. These were necessary to verify furnace performance prior to processing of actual material samples. Graphs of other flight system parameters were produced at various times during the mission, as well.

POSTMISSION DATA UTILIZATION

Virtually all forms of data that were archived during the mission have since been used to analyze different aspects of the mission. The hand written logbooks have been used to make initial estimates of total flight operating time, total sample processing time, total functional test time, and total Capaciflector experiment time. Entries in these logbooks have also been used as guides for sifting through the hundreds of archive data files.

The symbolic command log files have been used to identify the time when each material sample was processed. This time ordered list was then used as a guide for extracting data from the telemetry archives to produce graphs of furnace power versus time for every material sample and sample temperature versus time for every calibration sample. These graphs are now being used to analyze system performance during the flight as compared to pre- and post-flight test performance - the ultimate goal being to determine if any correlations can be made to the samples that were processed on-orbit. In the case of three sample runs, data from the backup archives were used to produce the graphs because the primary archive files were incomplete due to intermittent problems with the primary system during the mission.

In addition to analyzing furnace performance, archive data has also been used for evaluation of on-orbit robot performance and experiment thermal performance (ref. 5).

KEY ASPECTS OF ROMPS DATA SYSTEM DEVELOPMENT AND OPERATIONS

1) Rapid Telemetry and Command System Development

A notable aspect of this complex NASA sponsored commercially oriented experiment was its rapid pace of development: critical design review to shuttle integration took less than 18 months. The novel methodology of arranging system requirements by function and use of commercial hardware and software allowed fast prototyping and data system development.

2) Highly Flexible Control System

The ROMPS multilevel control system provided selectable degrees of autonomy - from highly interactive to totally autonomous. The principal investigators' ability to interact with their experiment in real time from the ground was a breakthrough in the ROMPS experiment.

3) Ground Station Based On Local Network of Macintosh Computers

Use of the Macintosh computer's fastest interface available, Ethernet instead of LocalTalk, in a distributed ground system computer network resolved the largest part of inter-system performance throughput problems.

4) Encoding Of HH Command Issuance Timing Separation Formula

Encoding of the HH format command issuance timing separation formula for uplink of long command sequences was needed to avoid HH Carrier command system overflow. During the mission, it was used to upload 10 Mbytes of the temperature-profile parameters and schedules. No command system overflows occurred.

5) Archive Telemetry Stream Delineated With HH Format Archived Commands

The archive file of real time telemetry frames was also archiving uplinked commands in hexadecimal format. This archive of commands was in addition to the symbolic command archive. This facilitated the telemetry analysis of what happened as a result of one command issuance until the next command issuance. This form of command archiving was also useful because archived commands could be displayed in hexadecimal form and used to debug the HH format commands during the system development phase of the ROMPS project.

6) Leasing of Backup Ground System Computers

The backup ground system computers were leased to reduce cost and provide computer delivery in a few days. It allowed the use of a separate system for I&T and prevented the break up of the primary system that was already at the POCC.

7) ROMPS Data System Hardware And Software Reusability

In addition to using the ROMPS data system to re-fly the ROMPS experiment, the system might also be reused to support other GSFC Hitchhiker payloads due to its highly flexible design and proven performance.

LESSONS LEARNED

Lesson 1- There is always at least one more customer on the mission besides you, namely the HH Carrier itself. HH Carrier designates three ID's for the HH Avionics commands issued by the carrier ground system. Because of this, you must always negotiate early with the HH Project for your needs and priorities, even when you think you are the only customer on your mission. All other non-Hitchhiker payloads during the mission also share with you the command channel and the low rate telemetry downlink, and compete for its bandwidth.

Lesson 2- CCGSE Command and Link packets are very useful during customer system development and I&T to verify the CGSE/CCGSE command and telemetry interface. These are not often used with a price paid in time lost during I&T and interface debugging.

Lesson 3- Verify that all advertised HH Carrier features are provided in your mission and use them whenever possible for development and I&T benefits.

Lesson 4- In planning your system development, plan for at least 5 software deliveries in increments of complexity. There will be a reduction in the number of change requests following acceptance testing of each delivery. This will insure that you are not overwhelmed at the last moment.

Lesson 5- Plan for contingencies during the mission by leaving some of your designated command windows empty of commands.

Lesson 6- Commercial software is not generally tested to the degree that custom designed space flight software is. Plan to diligently test every single feature and parameter of commercial software in the configuration planned for flight and then do not upgrade to newer versions. Beware of incompatibility between commercial packages. Remember that testing can show presence of errors but not absence of errors.

Lesson 7- Throughput limitation is a common problem. Do the most to insure that your ground system can handle the required data throughput multiplied by at least a factor of 2.

Lesson 8- Expect downlink communication channel data errors that cause short data dropouts during POCC operations that may not be recoverable. Generally, such short data dropouts (those less than a few minutes) are not treated as gaps and cannot be counted on being recovered in postmission playback requests. Such dropouts can wreak havoc with time scales when graphing flight data using commercial spreadsheet programs. Plan to either preprocess the data to restore lost time points or else devise a method to identify regions of missing data in the graphs.

Lesson 9- Expect some duplicate time stamps on successive one second telemetry frames. This is due to different frames occurring within the same second boundary and receiving the same time stamp. This is especially true of time stamps issued within the user's ground station due to the irregular flow of telemetry from the CCGSE.

Lesson 10- Your command channel throughput does not mean that you can continuously pump commands at the assigned channel rate. It just means that such a channel is there. Expect additional throughput limitations for non-sparse commanding.

Lesson 11- Sometimes there is a need to upload long serial asynchronous command sequences. For such cases, carefully plan the command sequence timing and inter-command delay to insure that the commands will not overrun the CCGSE command buffers and other HH carrier elements. Load as much as you can pre-flight into the flight system's nonvolatile memory because it can be very difficult to upload them at the POCC during the mission. It is likely you will run out of a command window or an error will occur during transmission, forcing you to repeat the upload, perhaps many times.

Lesson 12- Plan to provide the Hitchhiker project with the full commanding profile for your nominal mission. Within the profile, account for the time required for your payload to execute each command. This will force you to fully plan your mission and help to avoid last minute command design errors. In addition, KSC will require hexadecimal codes for all commands planned for the CITE test during shuttle integration. Test these codes repeatedly with your flight and ground system prior to providing the codes, and again prior to shipment of your payload to KSC. This will assure no codes have changed during late software modifications, thereby avoiding embarrassing and frustrating delays during shuttle integration testing.

Lesson 13- Operations simulations and training sessions are designed to insure you are prepared for mission operations. Do not trivialize their value and do not ignore or underestimate their significance. It will be much easier for you at the POCC during mission operations if you take full advantage of such events.

Lesson 14- Operations personnel at the POCC have participated in many missions while this is probably your first, and perhaps the last time, at the POCC. Listen to the advice of Operations personnel very carefully and attempt to follow it.

Lesson 15- At times your GSE will be required to support different activities at different sites - integration and testing at I&T facilities, simulations at the POCC, and shuttle integration tests at KSC. If possible, develop two ground systems, primary and backup, and use the backup system at all remote locations. This will avoid needless wear and tear on the primary system, eliminate worries over breaking up the primary system once it is at the POCC and tested, and provide a backup system during mission operations. Leasing the backup system may be attractive in terms of both cost and schedule.

Lesson 16- Sometimes advanced hardware features, like a computer's 32-bit addressing mode, cannot be fully utilized because the commercially delivered software was developed on an older system and does not run unless the advanced feature is turned off. Make sure that commercial software does not require superseded modes of operation and thus force you to underutilized the resources you paid for, like computer memory.

Lesson 17- Some basic tools like a telemetry frame generator must be required as part of a system delivery. A telemetry frame generator can be a hardware telemetry generator in a flight prototype board with a software module supplying telemetry source values or a ground based software telemetry frame generator.

Lesson 18- In ground systems based on networked computers with local databases, think in advance how you will insure that all databases remain synchronized, especially if change-only telemetry is being sent to downstream computers. Similarly, think about how you will set your GSE computers' clock to the POCC panel GMT time.

Lesson 19- Verify that time counter registers are implemented in sufficient size to avoid a rollover condition. I&T tests may not discover an error because the I&T activities are shorter than actual mission processing schedules.

Lesson 20- It is necessary to preserve the previous mission configuration for a re-fly mission and only change to accommodate new requirements. The environment present at the time of ROMPS1 mission completion in September of 1994 must be preserved. Commercial vendors must be notified to save and deliver to GSFC the system development environment and latest source code version. This will be needed to re-fly the ROMPS payload. All computer configurations must be frozen and no upgrades allowed for a mission re-fly unless new requirements emerge.

Lesson 21- The ROMPS mission resulted in a large amount of data to be processed and analyzed. Consider how you will process all of you data post-flight. The mission is not over until all necessary information has been extracted and sufficiently analyzed. Do not underestimate the amount of work involved in this. An object oriented data base for data products generation may be appropriate.

Lesson 22- In addition to automatically archived data, organize a method for all operations personnel to maintain written logs of events that occur during the mission. Insure that the time of each entry is listed. These logs will be of great assistance in evaluating data post-flight, especially when anomalies occur.

RESULTS AND CONCLUSIONS

ROMPS was one of the most complex Hitchhiker payloads flown to date. The mission was very successful with all of its goals fully attained. Its data system was based on a number of commercial products that resulted in a highly capable and flexible control system that provided selectable degrees of automation and graphical presentation of telemetry at the ground station. A number of very good lessons were learned while developing and operating the ROMPS experiment and these have been shared above.

TRADEMARK NOTICES

Spacecraft Command Language is a registered trademark of Interface and Control Systems, Inc. Easylab and System V are registered trademarks of Zymark Corporation.

Apple, LocalTalk, Macintosh, and Macintosh Quadra are registered trademarks of Apple Corporation.

LabVIEW is a registered trademark of National Instruments Corporation.

Bernoulli is a registered trademark of Iomega Corporation.

REFERENCES

1. Voellmer, G. M., "The ROMPS Robot in Hitchhiker", Shuttle Small Payloads Symposium, October 22, 1992
2. Olsztyn, P., Dobbs, M., Dickerhoof, G., Dorrance, P., "A Space-Based Laboratory Automation Architecture for Material Processing Experimentation", National Symposium on Laboratory Automation and Robotics, October 17-20, 1993, Boston, MA
3. Olsztyn, P., Dobbs, M., Conrad, D., "A Distributed Architecture For On-Orbit Laboratory Automation And Robotics Using COTS Components", AIAA 94-4506, 1994 AIAA Space Programs and Technologies Conference
4. *The Spacecraft Command Language Software System* , Interface & Control Systems, Inc., 1993
5. Bugby, D. C., "ROMPS Thermal Analysis Final Report", Swales & Associates Inc., Technical Report No. SAI-RPT-015, October 15, 1994
6. *Reliability, Maintainability, and Quality Assurance Manual*, NHB 5300.4, NASA Headquarters, Washington, DC

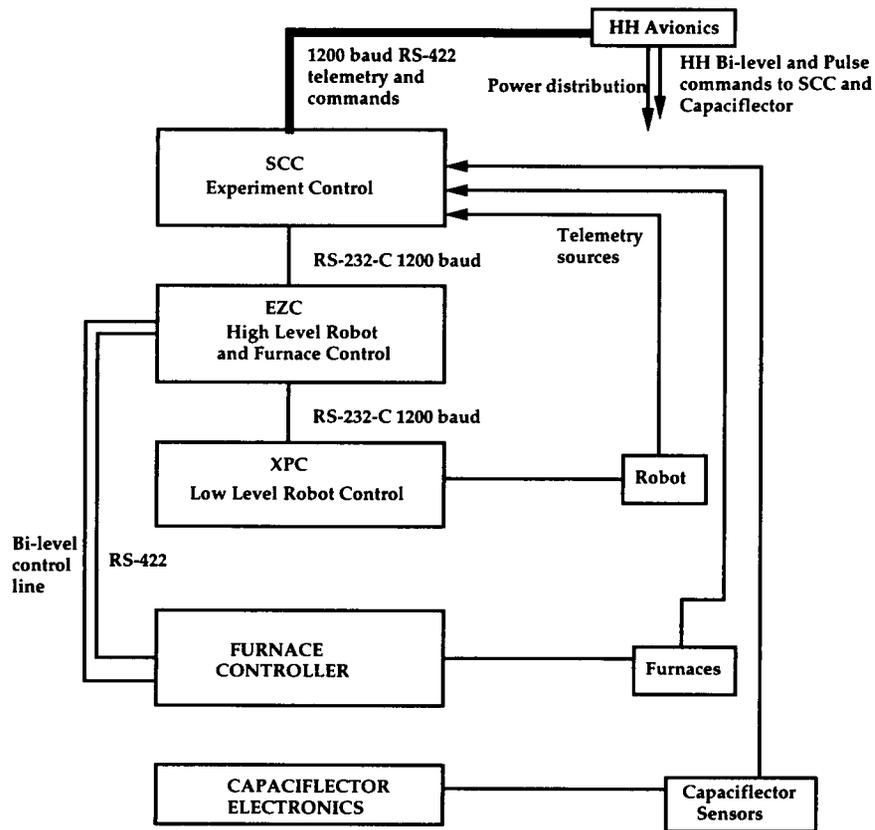


Figure 1: ROMPS Flight Data System Diagram

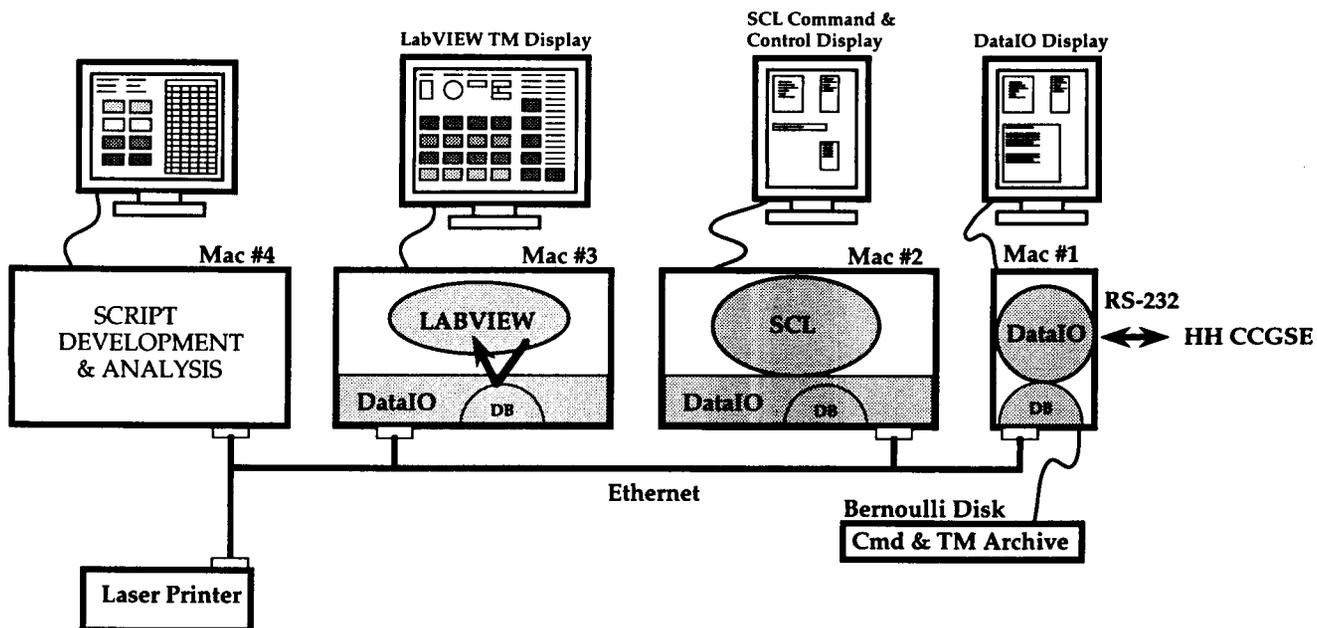


Figure 2: ROMPS Ground Station Hardware and Software